This project predicts the appearance of three Pokémon species using latitude and longitude data, exploring neural networks of varying complexity. The model achieves ~73% accuracy by capturing geographic patterns, with potential improvements from additional environmental and temporal features.

**Project Summary: Pokémon Go Classification**

**Objective:**  
Predict which of three Pokémon (Diglett, Seel, Tauros) is likely to appear at a given location based on latitude and longitude.

**1. Data Loading and Subsetting**

* Dataset: 296,021 Pokémon sightings with 208 features.
* Focused features: latitude, longitude, and class (mapped to Pokémon name).
* Filtered to three Pokémon → 2,083 instances: Diglett, Seel, Tauros.

**Key insight:** Only location features are used to predict Pokémon appearances, ignoring other factors like weather or time.

**2. Data Visualization**

* Class distribution roughly balanced (~33% each) → prevents bias.
* Scatter plots of longitude vs latitude reveal approximate separability:
  + Tauros in US, Seel in Europe, Diglett in Mexico and southern regions.
* Exact separation is difficult due to overlapping regions.

**Expected difficulty:**

* Coarse separation gives ~66% accuracy easily.
* Modeling exact distributions is harder due to overlaps.

**3. Preprocessing**

* Inputs: latitude & longitude; Outputs: class integers (Diglett→0, Seel→1, Tauros→2).
* Data split: Training 55%, Validation 25%, Test 20%.
* Custom dataset normalizes features using training mean/std and supports batching.

**Key points:**

* Stratified splitting ensures balanced classes.
* Normalization prevents scale issues for neural networks.

**4. Training and Model Capacity**

**Training Loop:**

* Forward pass → Loss (Cross-Entropy) → Backward pass → Parameter update (Adam).
* Accuracy tracked per batch and epoch; sanity checks included overfitting a single batch.

**Neural Network Architectures:**

* **SimpleNN:** 2 hidden layers, ReLU, Softmax → ~73% accuracy; captures geographical patterns.
* **TinyNN:** 1 linear layer → ~55% accuracy; coarse but generalizes continental patterns.
* **LargeNN:** 4 layers × 1024 units → high capacity, slower convergence, shows overfitting.

**Decision Regions:**

* SimpleNN captures intuitive patterns.
* TinyNN produces coarse, general regions.
* LargeNN overfits; visualization shows importance of sufficient data.

**Practical takeaway:**

* For catching all three Pokémon, focus on US.
* Intermediate networks may help fine-grained targeting; overly large models risk memorizing data.

**5. Evaluation**

* Test accuracy: ~73% (slightly higher than validation due to chance in splits).
* Validation set tunes model; test set assesses generalization.

**Confusion matrix insights:**

* Misclassifications relatively even (Class 1-2:31, 1-3:41, 2-3:37).
* Per-class accuracy: Class 1: 77%, Class 2: 81%, Class 3 (Tauros): 65% (rarer, geographically limited).

**6. Exploration**

**Travel strategy:**

* Target regions where all three Pokémon appear.
* Focus predictions on land regions; decision regions help identify feasible travel areas.

**Potential feature improvements:**

* Time of year, time of day, day of week, terrain type, weather, population density.
* Histograms show:
  + Diglett → Saturday nights
  + Seel → Thursday nights
  + Tauros → Friday nights
* All three prefer night and terrain type 13 → including these features can improve practical planning.

**7. Questions and Answers**

1. **Why use only latitude and longitude instead of all 208 features?**
   * The task focuses on predicting location-based appearances; other features may add noise.
2. **Why check if the dataset is balanced?**
   * To prevent bias toward majority classes and ensure fair learning.
3. **How does stratified splitting improve training?**
   * Ensures each subset reflects overall class distribution.
4. **Why normalize latitude and longitude using training statistics?**
   * Prevents scale issues, helping neural networks converge properly.
5. **What challenges do overlapping Pokémon locations introduce?**
   * Makes exact classification harder; reduces maximum achievable accuracy.
6. **Why choose a batch size of 128?**
   * Balances gradient stability and computational efficiency; smaller batches increase noise, larger batches may converge slower.
7. **Why compute training and validation losses separately?**
   * To monitor overfitting and generalization.
8. **How does overfitting a single batch serve as a sanity check?**
   * Confirms the network can learn simple patterns and gradients propagate correctly.
9. **Why does TinyNN generalize better than LargeNN despite lower training accuracy?**
   * Smaller capacity avoids overfitting; captures coarse patterns rather than memorizing noise.
10. **How does model capacity influence decision regions?**
    * Determines granularity: Tiny → coarse, Simple → balanced, Large → overfitted.
11. **Why is visualization of decision regions important?**
    * Helps understand model behavior and geographic generalization.
12. **How would adding more worldwide sightings affect decision regions and model choice?**
    * More data improves generalization; allows larger models without overfitting.
13. **Why might large models require normalized inputs?**
    * Prevents unstable gradients and accelerates convergence.
14. **Pros and cons of regression vs classification for Pokémon locations:**
    * **Pros:** Continuous outputs for precise location, flexible predictions.
    * **Cons:** Loses simple categorization; error interpretation more complex.